

WHAT IS CLAIMED IS:

1. A method of managing task execution in a storage system comprising:
measuring a parameter indicative of storage system workload; and
assigning priority of tasks executable on the system based on the measured
parameter.
2. The method according to Claim 1 further comprising:
counting a number of host input/output operations per unit time as the parameter
indicative of workload.
3. The method according to Claim 1 further comprising:
measuring interface bandwidth as a proportion of bandwidth capacity as the
parameter indicative of workload.
4. The method according to Claim 1 further comprising:
assigning to a task a maximum allowable utilization value at which the task is
authorized to execute;
maintaining a queue of tasks, the individual tasks having the assigned maximum
allowable utilization values;
measuring a current utilization value;
querying the tasks on the queue in the queue order;
executing a queried task that has an assigned maximum allowable utilization value
higher than the current utilization value; and
deferring to a next task on the queue, if any, for a queried task that has an assigned
maximum allowable utilization value lesser than the current utilization
value.
5. The method according to Claim 1 further comprising:
maintaining a plurality of task queues including a task queue that bases execution
on assigned and measured utilization, and at least one task queue with a
priority that differs from the utilization based queue.

6. The method according to Claim 1 further comprising:
maintaining a high priority task queue for queuing and executing, in the queue order, tasks assigned a high priority; and
maintaining a utilization task queue for queuing and executing tasks, when the high priority queue is empty, in an order based in part on the order of queuing and in part on assigned allowable utilization value of a task and a measured current utilization value.
7. The method according to Claim 1 further comprising:
maintaining a data structure associated with a utilization task queue indicative of allowable utilization of all tasks on the queue; and
executing or deferring execution of all tasks on the utilization task queue based on the data structure and a measurement of current utilization.
8. An array controller comprising:
an interface capable of coupling to a storage array;
a control logic; and
a code executable on the control logic comprising:
a performance measurement utility that measures a parameter indicative of storage array workload; and
a task management utility that assigns priority of tasks executable on the storage array based on the measured parameter.
9. The array controller according to Claim 8 wherein:
the executable code further comprises a queuing utility that maintains a task queue and processes the tasks based at least in part on a current measurement of storage array workload.

10. The array controller according to Claim 9 wherein the task management utility operates in combination with the queuing utility and the performance measurement utility to maintain a queue of tasks with each task assigned a threshold utilization, periodically measure current utilization, and execute tasks on the queue in the queue order so long as the current utilization meets the task threshold utilization.

11. The array controller according to Claim 8 wherein the performance measurement utility measures a performance criterion selected from among a group consisting of number of host input/output operations per unit time, interface bandwidth as a proportion of bandwidth capacity, disk busy, disk transfers per second, kbyte throughput per second, number of input/output operations per time interval, and input/output wait percentage.

12. The array controller according to Claim 8 wherein:
the executable code further comprises a queuing utility that maintains a plurality of task queues including a task queue that bases execution on assigned and measured utilization, and at least one task queue with a priority that differs from the utilization based queue.

13. The array controller according to Claim 8 wherein:
the executable code further comprises a queuing utility that maintains a high priority task queue for queuing and executing, in the queue order, tasks assigned a high priority, and that maintains a utilization task queue for queuing and executing tasks, when the high priority queue is empty, in an order based in part on the order of queuing and in part on assigned allowable utilization value of a task and a measured current utilization value of the storage array.

14. The array controller according to Claim 8 wherein:
the executable code further comprises a queuing utility that maintains a data structure associated with a utilization task queue indicative of allowable utilization of all tasks on the queue, and that executes or defers execution

of all tasks on the utilization task queue based on the data structure and a measurement of current utilization.

15. The array controller according to Claim 8 wherein:

the storage array is a Redundant Array of Independent Disks (RAID) array in a structure selected from among RAID0, RAID1, RAID2, RAID3, RAID4, RAID5, RAID6, RAID7, and RAID10.

16. An array controller comprising:

an interface capable of coupling to a storage array;

a control logic; and

a code executable on the control logic comprising:

a performance measurement utility that measures a parameter indicative of storage array workload;

a queue manager that maintains a task queue of tasks assigned a workload threshold value; and

a task management utility that executes tasks acting on the storage array with a priority based on the storage array workload parameter and order on the task queue.

17. The array controller according to Claim 16 wherein the queue manager maintains a high priority task queue for queuing and executing, in the queue order, tasks assigned a high priority, and that maintains a utilization task queue for queuing and executing tasks, when the high priority queue is empty, in an order based in part on the order of queuing and in part on assigned allowable utilization value of a task and a measured current utilization value of the storage array.

18. The array controller according to Claim 16 wherein the performance measurement utility measures a performance criterion selected from among a group consisting of number of host input/output operations per unit time, interface bandwidth as a proportion of bandwidth capacity, disk busy, disk transfers per second, kbyte throughput per second, number of input/output operations per time interval, and input/output wait percentage.

19. An article of manufacture comprising:

a controller usable medium having a computable readable program code embodied therein for managing task execution in a storage array comprising, the computable readable program code further comprising:
a code capable of causing the controller to measure a parameter indicative of storage array workload;
a code capable of causing the controller to assign priority of tasks executable on the storage array based on the measured parameter.

20. The article of manufacture according to Claim 19 wherein the computable readable program code further comprises:

a code capable of causing the controller to maintain a queue of tasks with each task assigned a threshold utilization;
a code capable of causing the controller to periodically measure current utilization; and
a code capable of causing the controller to execute tasks on the queue in the queue order so long as the current utilization meets the task threshold utilization.